|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 사업명 | 빅데이터 기반 AI 개발 전문가 양성 과정 | | | |
| 소속 | 조이름 | Team\_3 | 조장 | 김동욱 |
| 조원 | 이름 | 박재현 | 이름 | 이보윤 |
| 이름 | 조재경 | 이름 | 최서윤 |

|  |  |
| --- | --- |
| 주제명 | 클라이언트 / 서버 파일전송 프로그램 개발 |
| 요구사항 정의 | 클라이언트  1. rename으로 이미지 이름 변경(년월일시\_이미지이름\_경로)  2. rename후 socket으로 서버에 send  3. 서버응답 “ok”받기  4. 해당 soket으로 보낸 경로에 이미지 저장하기 (가급적 년월일시분)으로  5. 저장 후 DB에 년월일시분초 이름 경로 저장  6. 서버에서 보내는 DATA 대기  7. 서버에서 필요한 이미지 요청이 들어오면(경로와 이름이 들어오면 문자열 인덱스로 그 경로 찾아가서) 이미지를 FTP로 전송  서버  1. 클라이언트 요청 대기  2. 클라이언트에서 socket으로 데이터 전송하면 구분하여 DB에 저장  3. 저장 완료 “OK”전송  4. 서버관리자(권한자)가 LIKE로 이미지 이름 또는 년월일시분초로 검색을 클라이언트에게 보내기 보낼 때는 반드시 경로와 이름 또는 경로와 년월일시로 보내기  5. 클라이언트에서 보내오는 이미지를 FTP CWD경로에 저장하고 DB에 남기기  6. 로그남기기 |
| 일정 | 2024-02-14 ~ 2024-02-15 (2일)  - Socket 통신 확인 및 데이터베이스 저장 (1일)  - 이미지 요청 및 응답, 로그 남기기 구현, 문서작성 (1일) |
| 역할분담 | 개발  서버: 박재현, 최서윤  클라이언트: 김동욱, 이보윤, 조재경 |
| 개발환경 | 언어: Python  프레임워크: X  라이브러리: socket, ftplib, sqlite3  DataBase: SQLite  IDE: Visual Studio Code |
| 개발자원 | PC  cpu: i7  RAM: 16G  GPU: gtx1650  HDD: 1T |
| 프로그램 개발 배경 | 1. 데이터 공유의 필요성  - 조직, 개인 등 데이터의 효율적 공유와 관리 중요성 대두  - 파일 전송 프로그램 사용으로 사용자 간 신속한 데이터 전송 및 공유 가능  2. 원격 작업 증가  - 원격 작업 및 협업 증가에 따라 파일 전송 필수적  - 다양한 지리적 위치에 있는 사용자들 간 파일 전송 및 공유 가능한 도구 필요  3. 보안 요구  - 민감한 정보를 포함한 파일의 안전한 전송을 위한 데이터 보안 필요.  4. 효율적인 파일 관리  - 대용량 파일 전송 및 관리의 어려움  - 대용량 파일의 효율적 관리와 전송 가능한 기능 필요  5. 사용자 경험 향상:  - 쉽게 파일을 전송하고 관리할 수 있는 직관적인 인터페이스 제공의 중요성  - 사용자의 편리성 및 만족도 향상 |
| 개발 과정 | 클라이언트 구현:  - Python의 Socket 라이브러리를 사용하여 클라이언트 구현  - 서버로 파일 업로드, 서버로부터 파일 요청 및 다운로드 기능 구현  서버 구현:  - 파일 업로드 및 DB저장 가능한 서버 구현  테스트 및 디버깅:  - 프로그램의 안정성을 위해 다양한 시나리오에서 테스트 진행  - 발생한 버그 및 오류 해결 |
| 기능 구현 | 1. 클라이언트는 이름을 변경하고 변경한 경로를 서버에 소켓 통신을 통해 알려줍니다.    2. 소켓으로 보낸 경로를 DB에 경로, 이름, 날짜 순으로 저장합니다    3. 서버는 클라이언트가 보낸 이미지의 경로를 데이터베이스에 저장합니다.    4. 서버는 클라이언트에게 소켓 통신을 통해 원하는 이미지 파일 인덱스를 보냅니다.    5. 클라이언트는 DB에 SQL을 통해 파일을 찾고 파일을 FTP로 파일을 전송할 수 있도록 합니다.    6. 서버는 전송 받은 파일을 제 3자에게 FTP 통신을 통해 파일을 전달합니다.  7. 전달 후 로그 파일을 만듭니다. |
| 결과 및 평가 | 파일전송 서버-클라이언트 프로그램  1. 기본적인 파일 전송 기능 제공, 안정성과 보안을 고려하여 개발  2. 다중 클라이언트 지원으로 여러 사용자 동시 파일 전송 가능  3. 테스트 및 디버깅을 통한 안정성 확보  4. 계속적인 사용자 피드백 수렴 / 프로그램 기능 개선 및 보완 예정 |